Claims

WHAT IS CLAIMED IS:

1. A method for rotating an image for display on a display device comprising:

(a) receiving a command to rotate a source image located in off-screen memory;

(b) defining the source image as a texture;

(c) calculating the vertices of a rotated destination area; and

(d) mapping the source image as a texture into the rotated destination area.

2. The method of claim 1 including the step of tesselating the source image into a plurality of primitive vertices.

3. The method of claim 1 including the step of receiving one of user selected screen orientations that includes at least one of a 90 degree, 180 degree and 270 degree orientation.

4. The method of claim 1 including the step of using a different instruction set for each of a plurality of selected screen orientations to initiate the step of calculating the primitive vertices of the rotated destination area.

5. The method of claim 2 wherein the primitive vertices are one of triangle and rectangle vertices such that 3D rendering with texture mapping is used to rotate the source image based on the received rotation command.

6. The method of claim 1 including the steps of:

storing the source image as a bit map image in off- screen memory;

calculating the rotated vertex information for use in 3D rendering;

mapping source image as a texture into display memory; and

storing the rotated image into display memory.

7. A method for rotating an image for display on a display device comprising:

(a) receiving, through a driver, a command to rotate a source image;

(b) receiving and storing, by a driver, the source image in off-screen memory;

(c) calculating, by a driver, the primitive vertices of the rotated destination area; and

(d) mapping, using the 3D engine, the source image as a texture into rotated destination area.

8. The method of claim 7 including the step of receiving one of user selected screen orientations that includes at least one of a 90 degree, 180 degree and 270 degree orientation.

9. The method of claim 7 including the step of using a different instruction set for each of a plurality of selected screen orientations to initiate the step of calculating the primitive vertices of the rotated destination area.

10. The method of claim 7 wherein the primitive vertices are one of triangle and rectangle vertices such that 3D rendering with texture mapping is used to rotate the source image based on the received rotation command.

11. The method of claim 7 including the steps of:

storing the source image as a bit map image in off- screen memory;

calculating the rotated vertex information for use in 3D rendering;

mapping source image as a texture into display memory; and

storing the rotated image into display memory.

12. The method of claim 7 including the step of tesselating the source image into a plurality of primitive vertices.

13. A storage medium comprising:

memory containing executable instructions that when read by one or more processing units, causes the one or more processing units to:

provide a user interface to select a screen rotation angle;

receive a command to rotate a source image; and

define a coordinate transformation to calculate the rotated destination for mapping of the source image to facilitate rotation of the source image using a 3D rendering engine.

14. The storage medium of claim 13 including memory containing executable instructions that when read by one or more processing units, causes the one or more processing units to receive one of user selected screen orientations that includes at least one of a 90 degree, 180 degree and 270 degree orientation.

15. The storage medium of claim 13 containing executable instructions that when read by one or more processing units, causes the one or more processing units to use a different instruction set for each of a plurality of selected screen orientations to initiate defining the vertex coordinate transformation for the mapped source image.

16. The storage medium of claim 13 including memory containing executable instructions that when read by one or more processing units, causes the one or more processing units to tessellate the source image to determine the primitive vertices.